| amblem | **ÇANKAYA UNIVERSITY**  **Software Engineering Department** | A circular logo with colorful arrows  Description automatically generated |
| --- | --- | --- |

**SENG 491 – 492 Graduation Project**

**Software Requirements**

**Specification**

**SenseAI**

**Ece Heval Ünal**

**Mustafa Tozman**

**Özge Alkan**

**Sadrettin Anıl Karaçay**

**Zeliha Aybüke Baştürk**

**Version 1.1**

Table of Contents

[1](#_heading=h.gjdgxs) Introduction 1

[1.1](#_heading=h.30j0zll) Purpose 1

[1.2](#_heading=h.1fob9te) Scope 1

[1.3](#_heading=h.3znysh7) Definitions, acronyms, and abbreviations 1

[1.4](#_heading=h.2et92p0) References 1

[1.5](#_heading=h.tyjcwt) Overview 1

[1.6](#_heading=h.3dy6vkm) Version History 1

[2](#_heading=h.1t3h5sf) Overall description 2

[2.1](#_heading=h.4d34og8) Product perspective 2

[2.1.1](#_heading=h.2s8eyo1) System interfaces 2

[2.1.2](#_heading=h.17dp8vu) User interfaces 2

[2.1.3](#_heading=h.3rdcrjn) Hardware interfaces 3

[2.1.4](#_heading=h.26in1rg) Software interfaces 3

[2.1.5](#_heading=h.lnxbz9) Communication interfaces 4

[2.1.6](#_heading=h.35nkun2) Memory constraints 4

[2.1.7](#_heading=h.1ksv4uv) Operations 4

[2.1.8](#_heading=h.44sinio) Site adaptation requirements 4

[2.2](#_heading=h.2jxsxqh) Product functions 4

[2.3](#_heading=h.z337ya) User characteristics 4

[2.4](#_heading=h.3j2qqm3) Constraints 5

[2.5](#_heading=h.1y810tw) Assumptions and dependencies 5

[2.6](#_heading=h.4i7ojhp) Apportioning of requirements 6

[3](#_heading=h.2xcytpi) Specific Requirements 6

[3.1](#_heading=h.1ci93xb) External Interface Requirements 6

[3.1.1](#_heading=h.3whwml4) User interfaces 7

[3.1.2](#_heading=h.2bn6wsx) Hardware interfaces 7

[3.1.3](#_heading=h.qsh70q) Software interfaces 7

[3.1.4](#_heading=h.3as4poj) Communications interfaces 7

[3.2](#_heading=h.1pxezwc) Functional Requirements 7

[3.2.1](#_heading=h.49x2ik5) SenseAI users 8

[3.2.2](#_heading=h.2p2csry) User class 2 10

[3.3](#_heading=h.147n2zr) Performance requirements 10

[3.4](#_heading=h.3o7alnk) Logical database requirements 11

[3.5](#_heading=h.23ckvvd) Design constraints 11

[3.5.1](#_heading=h.ihv636) Standards compliance 11

[3.6](#_heading=h.32hioqz) Software system attributes 11

[3.6.1](#_heading=h.1hmsyys) Reliability 12

[3.6.2](#_heading=h.41mghml) Availability 12

[3.6.3](#_heading=h.2grqrue) Security 12

[3.6.4](#_heading=h.vx1227) Maintainability 12

[3.6.5](#_heading=h.3fwokq0) Portability 12

[4](#_heading=h.1v1yuxt) References 13

# Introduction

## Purpose

This subsection should

1. Delineate the purpose of the SRS;
2. Specify the intended audience for the SRS.

## Scope

This subsection should

1. Identify the software product(s) to be produced by name (e.g., Host DBMS, Report Generator, etc.);
2. Explain what the software product(s) will, and, if necessary, will not do;
3. Describe the application of the software being specified, including relevant benefits, objectives, and goals;
4. Be consistent with similar statements in higher-level specifications (e.g., the system requirements specification), if they exist.

## Definitions, acronyms, and abbreviations

This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to one or more appendixes in the SRS or by reference to other documents.

## References

This subsection should

1. Provide a complete list of all documents referenced elsewhere in the SRS;
2. Identify each document by title, report number (if applicable), date, and publishing organization;
3. Specify the sources from which the references can be obtained.

If the list of references is too long, it should be provided at the end of the document as a separate section.

## Overview

This subsection should

1. Describe what the rest of the SRS contains;
2. Explain how the SRS is organized.

## Version History

Provide a table of table of changes to this document

| **Version No** | **Description of change** | **Date** |
| --- | --- | --- |
| 1.0 | Initial Release | 15.10.2024 |
| 1.1 | Corrections and additions have been made | 06.11.2024 |

# Overall description

## Product perspective

This project aims to develop a multi-modal psychology chatbot that analyzes users' emotional states and contexts and helps people with psychological guadline. This chatbot will perform emotional analysis using biometric data obtained from voice, facial expressions and wearable devices (e.g. smart watches) and provide personalized feedback. Users can use this application when they need instant support, check their old data and perform emotional state analysis. The application can be authenticated with the authentication API. Users' previous conversations will be deleted after a certain period of time in the database.

### System interfaces

**User Interface:** The interface through which users interact with the chatbot. This interface receives input from the user, such as voice, facial expressions, biometric data, and text input.

**Voice Recognition Interface:** Analyzes the user's voice commands and emotional tone.

**Image Processing Interface:** A system interface that analyzes the user's facial expressions.

**Biometric Data Interface:** An interface that analyzes the biometric data the user receives from wearable devices (e.g. smart watches).

**API Interfaces:** An interface that analyzes the data from the user in real time and provides instant feedback and suggestions by enabling the chatbot to integrate with external services (emotion analysis, biometric data collection, etc.). It is required for the client and server to communicate, allowing real-time data exchange and feedback. It ensures that requests sent by the client (user's device) are properly handled and processed by the server, which then returns appropriate responses to enrich the user experience.

**Database Interface:** An interface where user data is stored and queried.

### User interfaces

* The interfaces in the system should be designed to be user-friendly, guiding and easy to use.
* The login screen should be large enough to enter the username and password and should be adaptable to different screens. There should be no elements that will make it difficult for the user to click on a field or impair the readability of the interface.
* The buttons that the user will click on to select the desired data type on the homepage should be large enough and the fonts should be readable.
* The home screen should provide users with easy access to the new chat, previous chats and the user page.
* The menu structure should be intuitive so that users can easily access the information they need. The main menu should provide users with quick access to the desired section.
* Error messages should provide clear and direct information to the user.
* The interface should allow users to perform functions quickly and effectively.
* The steps required for any operation should be minimized. Unnecessary complexity should be avoided and a user should be able to easily use this application with an average of 5 minutes of training.
* The interfaces should be designed in a way that the user can easily remember how to use the application.
* The application shall provide support for **Dark Mode** on both Android and iOS platforms. Dark Mode should be available as an optional interface theme for users who prefer a darker, more visually comfortable interface. This feature will automatically respect the device's system-wide theme settings (if enabled) and allow the user to toggle between Light and Dark modes from within the app settings.
* The application shall ensure that the user interface (UI) remains consistent and responsive across different screen sizes, resolutions, and platforms (Android and iOS). The design will be adaptive, ensuring that the app provides an optimal and consistent user experience regardless of whether the user is on a phone, tablet, or other device.

### Hardware interfaces

This application is a mobile app and will be coded for Android. However, supported updates for IOS will be added later.

### Software interfaces

**Operating System (OS):** The application is built using the Flutter framework and relies on Android and iOS as the primary operating systems. The minimum required versions are Android 10 (API level 29) for Android devices and iOS 13.0 for iOS devices. These versions ensure that the application can access modern system APIs needed for biometric data, voice recognition, image processing, and other critical features. These OS versions also support improved privacy controls, security enhancements, and compatibility with Flutter plugins necessary for device functionalities.

**User Interface:** This is the interface where users interact with the chatbot. This interface receives various inputs from the user, such as voice, facial expressions, biometric data, and text input. It will be developed to have a user-friendly design, so that users can easily communicate with the chatbot.

**Voice Recognition Interface:** This is an interface that analyzes the user's voice commands and emotional tone. This interface ensures that voice commands are correctly perceived and that user emotions are extracted from the tone of voice. Real-time processing of the user's voice is a critical component for instant responses.

**Image Processing Interface:** This is a system interface that analyzes the user's facial expressions. This interface processes the necessary image data to determine the user's emotional state using face recognition and emotion analysis technologies. Real-time face recognition and emotion detection play an important role in the chatbot providing personalized feedback to the user.

**Biometric Data Interface(Google Fit):** An interface that analyzes biometric data received from wearable devices (e.g. smartwatches). This interface collects and interprets information such as heart rate, stress levels, and other health data. Integration of biometric data allows for more accurate feedback by providing a deeper understanding of the user's emotional state.

**API Interfaces:** An interface that analyzes data from users and provides feedback and recommendations. This interface is used to provide sentiment analysis, biometric data collection, and integration with other external services. API interfaces enhance the functionality of the chatbot, enriching the user experience. It is required for the client and server communication.

**Database Interface:** An interface where user data is stored and queried. This interface securely stores users' past sentiments, interactions, and other important information, and provides quick access when needed. The database interface is critical for managing and analyzing user data.

### Communication interfaces

The system will not communicate with any other application.

### Memory constraints

Users' chats will be kept in the database for a certain period of time. Then the data will be permanently deleted from the database.

### Operations

Users can register and log in to the system.

Users can choose what type of query to send (text, image, audio, and biometric data).

Users can change the data they entered when registering.

Users can access and delete previous chats.

The system shall support user authentication via an external authentication API to secure user accounts.

Users can start new chats at any time.

Users can choose the type of data to send before initiating the emotional analysis process.

The system will allow users to update or delete the information entered during registration.

The system shall provide error handling and feedback to the users when a problem arises (e.g., data format issues or network connectivity).

Users will be notified if they exceed the character limit for text inputs or try to upload unsupported files.

The application will guide users to ensure images are taken with sufficient lighting and at appropriate angles for accurate sentiment analysis.

The system will analyze and process multiple types of inputs (biometric, audio, image) and provide real-time emotional feedback based on those inputs.

Users will be able to view and analyze historical data through past chats.

The system will offer multilingual support in future versions, starting with English as the primary language.

The application will manage user data in compliance with privacy laws, ensuring secure storage and processing of sensitive information.

### Site adaptation requirementsThe application will be developed for both iOS and Android platforms, with Android as the primary target.

## Product functions

* The application will analyze the users' emotional state and provide real-time feedback.
* The application should comply with ethical standards while providing advice on mental health and should be sensitive to the needs of the users.
* Users can log in to the system by registering.
* Users can send text, image, audio and biometric data.
* The user should be able to choose what type of data to send from the homepage.
* Users can start new chats whenever they want and access and delete old chats from the user page.
* With the user-friendly interface, the user's interface will be easy to learn and remember with the use of the application.
* Wearable devices or existing health applications will be used for biometric data integration.
* The application will perform emotional analysis through multi-modal inputs such as voice, facial expressions, and biometric data collected from wearable devices like smartwatches.
* System will produce meaningful reports from the analysis.

## User characteristics

The intended users of SenseAI are primarily individuals seeking therapeutic assistance through a digital platform. Users are expected to have a general understanding of using mobile applications. No specific educational background is required; however, the app will be designed to cater to users with diverse educational levels, from high school students to professionals.

Given the nature of SenseAI as a mental support app using AI, users may have varying levels of comfort with AI interactions. Some users may feel hesitant or cautious about AI’s ability to understand emotional context. Consequently, the app will prioritize transparent communication about data handling, privacy, and the limitations of AI responses to build user trust and enhance user comfort.

## Constraints

* The size and image resolution of the videos that users will send will be determined later by the developer team and will have certain limits. Users who try to add data with an unsupported feature will be shown an appropriate warning message.
* The text received from users will have a character limit that will be determined later. Users who exceed the limit will be shown a warning message appropriate for the error.
* User data must be stored and processed in accordance with data privacy laws.
* Users' previous chats will continue to be kept in the database for a period to be determined during development.
* 95% of the responses returned to the user should be under 1 minute.
* In the first stage, 5 users will be supported simultaneously, and this number will be increased later according to hardware.
* The app size should be under 150 mb

## Assumptions and dependencies

**User permissions**: Permissions such as camera and microphone access are assumed to be allowed by the user.

**User Device Specifications:** The characteristics of the devices that users use to access the software can affect the performance of the application. For example, optimization requirements may arise for devices with lower hardware specifications. The mobile devices are assumed to have working camera microphone and touch screen. Also users are assumed to be familiar with basic smart device interactions.

**Network Conditions:** Users' internet connection speeds and network conditions can affect the performance of the software. Conditions such as longer response times or data loss for low-bandwidth connections may require reconsideration of the requirements.

**Audio and Visual Quality:** The audio and video quality delivered to users can significantly impact the user experience. Inadequate audio or video quality can make communication difficult to understand and can reduce the effectiveness of the application. Therefore, the requirements should be updated to ensure high-quality audio and video streaming.

## Apportioning of requirements

**Additional Emotion Recognition Modes:** Additional emotion recognition modes (e.g. stress management or relaxation techniques) added to better understand the user’s mood.

**More Integration Options:** A wider range of integrations with different wearable devices. For example, connection with more brands and models of smartwatches or fitness trackers.

**Enhanced UI Features:** Added more customization options in the UI design to enhance the user experience.

**Multilingual Support:** Added multilingual support features to enable the app to be used in different languages.

# Specific Requirements

This section of the SRS should contain all of the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. Throughout this section, every stated requirement should be externally perceivable by users, operators, or other external systems. These requirements should include at a minimum a description of every input (stimulus) into the system, every output (response) from the system, and all functions performed by the system in response to an input or in support of an output. As this is often the largest and most important part of the SRS, the following principles apply:

1. Specific requirements should be stated in conformance with all the characteristics described in 4.3 of the standards.
2. Specific requirements should be cross-referenced to earlier documents that relate.
3. All requirements should be uniquely identifiable.
4. Careful attention should be given to organizing the requirements to maximize readability.

## External Interface Requirements

This should be a detailed description of all inputs into and outputs from the software system. It should complement the interface descriptions in section 2 and should not repeat information there.

It should include both content and format as follows:

1. Name of item;
2. Description of purpose;
3. Source of input or destination of output;
4. Valid range, accuracy, and/or tolerance;
5. Units of measure;
6. Timing;
7. Relationships to other inputs/outputs;
8. Screen formats/organization;
9. Window formats/organization;
10. Data formats;
11. Command formats;
12. End messages.

### User interfaces

### Hardware interfaces

### Software interfaces

### Communications interfaces

## Functional Requirements

Functional requirements should define the fundamental actions that must take place in the software in accepting and processing the inputs and in processing and generating the outputs. These are generally listed as "shall" statements starting with "The system shall."

These include

1. Validity checks on the inputs
2. Exact sequence of operations
3. Responses to abnormal situations, including
   1. Overflow
   2. Communication facilities
   3. Error handling and recovery
4. Effect of parameters
5. Relationship of outputs to inputs, including
   1. Input/output sequences
   2. Formulas for input to output conversion

It may be appropriate to partition the functional requirements into subfunctions or subprocesses. This does not imply that the software design will also be partitioned that way.

1. The system shall be available to users 24/7.
2. The system shall analyze users' emotional states and provide feedback.
3. The system shall comply with ethical standards when providing mental health advice and shall be sensitive to users' needs.
4. The system shall initially support the English language
5. The system shall allow users to register and log in to the application.
6. The system shall enable users to send data via text (up to X characters), images, audio, and biometric data (BPM readings from smart watches).
7. Users shall be able to choose the type of data to submit before analysis.
8. The system shall retain users' previous inputs in the database for a period to be determined during development.
9. Users shall be able to start new chats and access or delete old chats from their history page.
10. The system shall allow users to update and delete the information entered during registration.
11. The system shall provide a user-friendly interface that is easy to learn and remember. The interface should also have an appealing color palette.
12. The system shall guide users to ensure images are taken in sufficient lighting and at appropriate angles to maintain data quality for accurate sentiment analysis.
13. The system shall clearly convey the problem to the users when an error occurs and allow recovery.
14. The system shall initially support the English language, with potential for multilingual expansion in the future.
15. The system shall support user authentication via an external authentication API to secure access to personal data and previous interactions, safeguarding user privacy.

### SenseAI users

#### Logging in

##### Introduction/Purpose of feature

The Logging In feature allows authorized users to securely access the SenseAI application and their personalized data. This feature serves as an entry point to ensure data privacy, personalized experience, and user accountability. The system should let users stay logged in once login is successfully completed.

##### Stimulus/Response sequence

1. System: Check if user logged in before
2. System: Direct to landing page if user logged in
3. System: Else, show login screen with text fields for e-mail and password
4. User: Fills in account info
5. User: Tap login
6. System: Check account credentials and direct user to landing page if correct
7. System: Else show error

##### Associated functional requirements

System: Check if user logged in before

System: Direct to landing page if user logged in

System: Else, show login screen with text fields for e-mail and password

User: Fills in account info

User: Tap login

System: Check account credentials and direct user to landing page if correct

System: Else show error

#### Sending text for analysis

##### Introduction/Purpose of feature

The system shall allow users to input text for emotion and sentiment analysis, enabling SenseAI to interpret emotional cues within user responses. By processing the input, SenseAI’s analysis module will detect underlying emotions (e.g., happiness, sadness, anger) and overall sentiment (positive, negative, neutral).

##### Stimulus/Response sequence

1. System: Show textbox for input
2. User: Enter input.
3. User: Taps "Send" button.
4. System: Validates text input and sends it to the model for analysis with other input types (if exists).
5. System: Displays loading indicator.
6. Model: Processes the input.
7. System: Receives analysis results, saves it to the database and the device.
8. System: Displays results to the user.

##### Associated functional requirements

#### Using voice as input

##### Introduction/Purpose of feature

* + - * 1. The system shall allow users to input voice by processing the audio through text-to-speech.

##### Stimulus/Response sequence

1. System: Show microphone icon for input
2. User: Taps the icon
3. User:Speaks to the microphone.
4. System: Detects speech and uses TTS (text to speech) and converts it to text input.
5. System: Validates input and sends it to the model for analysis with other input types (if exists).
6. System: Displays loading indicator.
7. Model: Processes the input.
8. System: Receives analysis results, saves it to the database and the device.
9. System: Displays results to the user.

##### Associated functional requirements

REQ 1- Transcript of the user’s input should be added to the input textbox, ready to be sent.

#### Sending image for analysis

##### Introduction/Purpose of feature

The system shall allow users to input photos using their smart devices for analysis, enabling SenseAI to process the images and extract meaningful information for better analysis.

##### Stimulus/Response sequence

1. System: Show image icon for input
2. User: Taps the icon.
3. System: Enables the front camera for taking a photo
4. User: Takes a photo.
5. System: Validates input and sends it to the model for analysis with other input types (if exists).
6. System: Displays loading indicator.
7. Model: Processes the input.
8. System: Receives analysis results, saves it to the database and the device.
9. System: Displays results to the user.

##### Associated functional requirements

REQ 1-Users should be able to take photos without exiting the app.

#### Sending heartbeat info for analysis

##### Introduction/Purpose of feature

The system shall enable SenseAI to receive and analyze users' heartbeat information to assess emotional or physical states. This analysis will take into account the user’s age, as age-related differences can impact heart rate norms and interpretations. By factoring in age, SenseAI can more accurately identify deviations from typical heartbeat patterns, helping to gauge stress, anxiety, or relaxation levels.

##### Stimulus/Response sequence

1. System: Display option for biometric data input.
2. User: Either check or uncheck the option
3. System**:** Check if the user has checked or unchecked the option for biometric data input. If checked, proceed; if unchecked, skip the biometric data retrieval step.
4. System**:** Check if Google Fit API is successfully connected to the user’s account (is the user authenticated on Google Fit?).
5. System**:** Check if the required biometric data (BPM) is available in Google Fit. If data is unavailable or incomplete, prompt the user to provide manual input or skip this step.
6. System: Access today’s google fit information(BPM) of the user and send it to the model along with other input type (if exists)
7. System: Displays loading indicator.
8. Model: Processes the input.
9. System: Receives analysis results, saves it to the database and the device.
10. System: Displays results to the user.

##### Associated functional requirements

REQ-1 The system should be able to access the latest heartbeat info from the user’s google fit account.

REQ-2 The users should be only able to use this input option if they have a google fit account linked. If no account is linked the option should be grayed out. And tapping it should show a text bubble telling users they need to link their google fit accounts

REQ-3 If an error occurs while retrieving info or if no data exists, the app should inform the user about the error

#### Viewing previous analysis (history)

##### Introduction/Purpose of feature

The system shall provide users with access to a history feature, allowing them to view previous analyses and insights generated by SenseAI. This history will include past sentiment, emotion analyses from text, heartbeat, and image inputs, enabling users to track trends and changes over time. By offering a historical view, SenseAI aims to support users in understanding their emotional patterns and progress

##### Stimulus/Response sequence

System**:** Display option to view past analysis results

User**:** Selects the option to view previous analysis.

System**:** Checks if the user has any stored analysis results in the database or device.

* If results are found, proceed.
* If no results are found, display a message such as "No previous analysis available."

System**:** Retrieves and displays a list of previous analysis results.

User**:** Selects a specific past analysis entry to view detailed results.

System**:** Displays the selected analysis details, including insights, graphs, and any other relevant data.

User**:** Optionally, chooses to delete or update past analysis results (if allowed by the system).

System**:** If the user deletes or updates, confirms action and updates the database accordingly.

System**:** displays the updated history of deletion.

##### Associated functional requirements

REQ 1- History of previous interactions should be displayed with dates.

REQ 2- Users should be able to view the history chronologically ordered.

REQ 3- Users should be able to delete previous talks.

REQ 4 - The system should provide a search or filter option, allowing users to find specific past interactions based on keywords or date ranges.

### Admin

.

.

.

## Performance requirements

1. The application shall be available 24/7 to ensure accessibility at any time.
2. 95% of system responses to user requests shall be completed within 1 minute to ensure timely feedback.
3. The system shall initially support up to 5 simultaneous users, with scalability to support up to 100 users as the application matures and a more advanced hardware is provided.
4. The system shall handle diverse data types, including text, images, audio, and biometric data.
5. The system shall be able to store user-generated content and chat history for a development-specified retention period, allowing for selective user-managed retention options.
6. Users who engage with audio and video functionalities shall use compatible devices to ensure smooth data transmission and processing.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

This subsection should specify both the static and the dynamic numerical requirements placed on the software or on human interaction with the software as a whole. Static numerical requirements may include the following:

1. The number of terminals to be supported;
2. The number of simultaneous users to be supported;
3. Amount and type of information to be handled.

Static numerical requirements are sometimes identified under a separate section entitled Capacity.

Dynamic numerical requirements may include, for example, the numbers of transactions and tasks and the amount of data to be processed within certain time periods for both normal and peak workload conditions.

All of these requirements should be stated in measurable terms. For example,

*95%* *o*f *the* *transactions* *shall* *be* *processed* *in* *less* *than* 1 *s.*

rather than,

*An* *operator* *shall* *not* *have* *to* *wait* *for* *the* *transaction* *to* *be completed.*

NOTE-Numerical limits applied to one specific function are normally specified as part of the processing subparagraph description of that function.

## Logical database requirements

This should specify the logical requirements for any information that is to be placed into a database. This may include the following:

1. Types of information used by various functions;
2. Frequency of use;
3. Accessing capabilities;
4. Data entities and their relationships;
5. Integrity constraints;
6. Data retention requirements. (Data will be backed up at the end of each day)

## Design constraints

This should specify design constraints that can be imposed by other standards, hardware limitations, etc.

The device must support camera, microphone and smart watch.

### Standards compliance

* + - 1. **Material Design Guidelines**

The user interface for Sense AI will follow Google’s Material Design Guidelines to provide a consistent, modern, and intuitive experience. Material Design offers a comprehensive set of visual, motion, and interaction design principles that enhance usability and aesthetic appeal.

This subsection should specify the requirements derived from existing standards or regulations. They may include the following:

1. Report format;
2. Data naming;
3. Accounting procedures;
4. Audit tracing.

For example, this could specify the requirement for software to trace processing activity. Such traces are needed for some applications to meet minimum regulatory or financial standards. An audit trace requirement may, for example, state that all changes to a payroll database must be recorded in a trace file with before and after values.

## Software system attributes

There are a number of attributes of software that can serve as requirements. It is important that required attributes be specified so that their achievement can be objectively verified.

### Reliability

This should specify the factors required to establish the required reliability of the software system at time of delivery.

Data will be backed up to prevent loss.

### Availability

This should specify the factors required to guarantee a defined availability level for the entire system such as checkpoint, recovery, and restart.

### Security

This should specify the factors that protect the software from accidental or malicious access, use, modification, destruction, or disclosure. Specific requirements in this area could include the need to

1. Utilize certain cryptographical techniques;
2. Keep specific log or history data sets;
3. Assign certain functions to different modules;
4. Restrict communications between some areas of the program;
5. Check data integrity for critical variables.

### Maintainability

This should specify attributes of software that relate to the ease of maintenance of the software itself. There may be some requirement for certain modularity, interfaces, complexity, etc. Requirements should not be placed here just because they are thought to be good design practices.

The application will be extendable since it will be developed in an object-oriented manner and will be modular.

### Portability

This should specify attributes of software that relate to the ease of porting the software to other host machines and/or operating systems. This may include the following:

1. Percentage of components with host-dependent code;
2. Percentage of code that is host dependent;
3. Use of a proven portable language;
4. Use of a particular compiler or language subset;
5. Use of a particular operating system.

# References

The resources used to produce this document should be listed here and referenced in the relevant text of this document

| [1] | Software Engineeering Department, "Graduation Projects," SENG, [Online]. Available: https://seng.cankaya.edu.tr/graduation-projects/. [Accessed 28 June 2024]. |
| --- | --- |